1. **What is Abstract Class and method explain with Example.**

**Abstract Class:**

* A class that consist of one or more abstract method is called as abstract class.
* In python Abstract class must be inherit to Abc class of abc module.
* If you inherit an abstract class, it must be implement all the abstract method of that class otherwise child class also become abstract class.
* Syntax:

From abc import ABC

class class\_name:

body of class

**Abstract Method:**

* An abstract method is which is declared without any body (implementation). i. e Empty method is known as abstract method.
* An abstract method is a method that has a declaration but does not have an implementation.
* It is also declared with @abstractmethod decorator of abc method.

**Example of Abstract Method and class.**

from abc import ABC, abstractmethod

class Shape(ABC):

@abstractmethod

def draw(self):

pass # Without implementation

@abstractmethod

def area(self):

pass

class Circle(Shape):

def draw(self):

print("Draw The Circle")

def Circumference(self):

return'Circumference of circle'

def area(self):

return 'Area of Circle.'

cir=Circle()

print(cir.Circumference())

cir.draw()

print(cir.area())

1. **What is Exception.**

Exception is abnormal condition which disturbs normal flow of program execution.

Due to an exception program is terminated abnormally.

Exception is also known as Error.

Example:-

Divide any number

10/0 --> infinity

Python interpreter is not able execute above statement then it give an exception.

i. e ZeroDivisionError

l1=[3,5,67,86]

print(l1[4])

here we got the IndexError.

There are 2 Types of error:-

1. Runtime Error:-

-This Error raise during the execution of program.

And terminate program abnormally.

but using exception handling we can handle them.

Exception

IndexError

ZeroDivisionError

ValueError

KeyError

FileNotFoundError

etc.

2. Syntax Error:-

-This error raise before execution of program

i. e at a time of program interpretation.

So we can't handle these using Exception Handling.

1. **What is Exception handling.**

Exception Handling is a mechanism in which we can handle the Runtime Error and avoid abnormal termination of program.

Because of Exception handling program execution will successfull completed with an error.

To handle Exception python we suppose to use some block or clause.

1. **Try block:-**

This is used to write the code which raise the exception during execution.

and threat exception/Error we easily except using except block.

1. **Except block:-**

This is used to write code which we want to execute after

exception is raised in try block.

(i. e it used to write handling code of error like error message printing act)

try:

Critical Statement

except Error\_class as obj:

code to execute...

1. **Else block:-**

This block is used to define a statement which we want to execute

without exception.

**4. Finally block**

It will always execute with or without exception.

Finally block can be define after try block as well as except block also.

**4. Why we need the exception Handling.**

* Exception handling is used to complete program successfully with an error.
* It is important because it helps to maintain the normal, desired flow of the program even when unexpected events occurs.
* If python exceptions are not handled, program may crash or requests may fail.
* If we do not handle the exception, the interpreter doesn’t execute all the code that exist after the exceptions.